

What is claimed is:

1. A method of correcting an exception during a printing process at least partially controlled by a plurality of print process modules associated with a printing device, the method comprising:

5 monitoring, from a self-correcting module, a state of each of a plurality of the print process modules, wherein the print process modules interact according to a set of rules to control the printing process;

determining that an event has occurred; and

setting a current state of the at least one print process module to a default

10 condition.

2. The method of claim 1, wherein determining that an event has occurred includes determining that the printing device is hung.

15 3. The method of claim 2, wherein determining that the printing device is hung is based on at least one predetermined rule and the state of at least one print process module

20 4. The method of claim 1, wherein monitoring includes receiving a status message from each of the printing process modules into a global event history queue of the self-correcting module

25 5. The method of claim 1, wherein determining includes examining the global event history queue to determine whether the conditions of the predetermined rule are met.

6. The method of claim 5, wherein determining further includes verifying that the conditions of the rule remain satisfied over a predetermined period of time.

7. The method of claim 1, wherein setting is accomplished at least in part by sending a reset command to the at least one print process module from the self-correcting module.

5 8. The method of claim 1, further comprising resending at least a portion of the print job to the at least one print process module.

9. The method of claim 3, wherein the plurality of print process modules includes a paper path module.

10

10. The method of claim 9, wherein the paper path module includes a print controller and an engine controller.

11. The method of claim 10, wherein the predetermined rule is:

15 if, for a predetermined period of time, a state of a current print job is processing, and a current state of the engine controller is ready, and a current state of the print controller is waiting for an associated print engine to be ready, then send a reset command to the print controller and send a reset command to the engine controller, to cause each of the print controller and the engine
20 controller to return to a default state.

12. The method of claim 10, wherein the predetermined rule is:

if, for a predetermined period of time, a current state of a current print job
25 is pending cancellation, and a current state of the engine controller is received cancel request, and a current state of the print controller is waiting for cancel reply, then sending a reset command to the print controller and sending a reset command to the engine controller, to cause each of the print controller and the engine controller to return to a default state.

30 13. The method of claim 1, wherein each of the print process modules is stored as firmware within the printing device.

14. A method of correcting an exception during a printing process in a printing device, comprising:

monitoring a current state of a plurality of print process modules in the printing device, the print process modules being configured to at least partially control the printing process;

determining an expected state of at least one print process module;

comparing the current state to an expected state of the at least one print process module;

detecting a discrepancy between the current state and the expected state;

10 and

setting the current state of the at least one print process module to a default condition.

15. The method of claim 14, wherein the plurality of print process

modules includes a job module, a paper path module, and a data path module.

16. The method of claim 14, wherein monitoring the current state includes receiving a status message from each of the print process modules at a self-correcting module.

20

17. The method of claim 16, wherein determining includes determining an expected state for the at least one print process module based on an event history and a predetermined event rules.

25

18. The method of claim 16, wherein the self-correcting module is stored at least partially as firmware of the printing device.

30

19. The method of claim 16, wherein the self-correcting module is stored at least partially within software in communication with the printing device through a network.

20. The method of claim 14, wherein the plurality of print process modules are stored at least partially as firmware in the printing device.

21. A self-correcting printing system comprising a printing device
5 having an instruction set including:

a plurality of print process modules configured to at least partially control a printing process in the printing device;

a self-correcting module including:

10 a module status monitor configured to monitor a current state of at least a plurality of print process modules of the instruction set;

a plurality of event rules that describe a manner in which the plurality of print modules interact;

15 an event history configured to store information relating to a state of the print process modules;

a hang detector configured to detect a hang condition among the plurality of print process modules based on the event history and event rules; and

20 a correction mechanism configured to change the current state of the at least one print process module to a default state, upon detection of the hang condition.

20

22. The printing system of claim 21, wherein the hang detector further includes a comparison mechanism configured to determine an expected state of the at least one print process module, based on the event history and event rules, and to detect a discrepancy between the expected state and the current state of 25 the at least one print process module.

23. The printing system of claim 21, wherein the plurality of print process modules includes a job module, a paper path module, and a data path module.

5 24. The printing system of claim 21, wherein the plurality of print process modules are stored as firmware on the printing device.

10 25. The printing system of claim 21, wherein the self-correcting module is stored at least partially within software in communication with the printing device through a network.

26. The printing system of claim 21, wherein the printing device further includes a laser print mechanism, configured to apply print to a print medium.

15 27. The printing system of claim 21, wherein the module status monitor is configured to monitor the current state of the print process modules by receiving status messages from each of the print process modules, and storing the status messages in the event history.

20 28. The printing system of claim 21, wherein the self-correcting module further includes a timer configured to determine whether conditions of a rule are satisfied for a predetermined period of time, and wherein the correction mechanism is configured to change the current state of the at least one print process module after the predetermined period of time has elapsed.

29. A computer program product comprising:

a computer usable medium having computer readable program code embodied therein for causing correction of an exception condition within firmware of a printing device, the computer readable program code in said computer program product comprising:

computer readable program code configured to cause the printing device to determine a current state of at least one module of the firmware;

computer readable program code configured to cause the printing device to compare the current state to an expected state;

10 computer readable program code configured to cause the printing device to detect a discrepancy between the current state and the expected state; and

computer readable program code configured to cause the printing device to set the current state to a default condition.

15 30. The computer program product of claim 29, further comprising computer readable program code configured to cause the printing device to determine the expected state from an event history.

31. The computer program product of claim 30, wherein the expected
20 state is determined from the event history using a set of event rules.